How to Publish ALEPH Records in Google

Version 18 and later

ExLibris The bridge to knowledge

Documentation produced June 2007
Web address: http://www.exlibrisgroup.com

Table of Contents

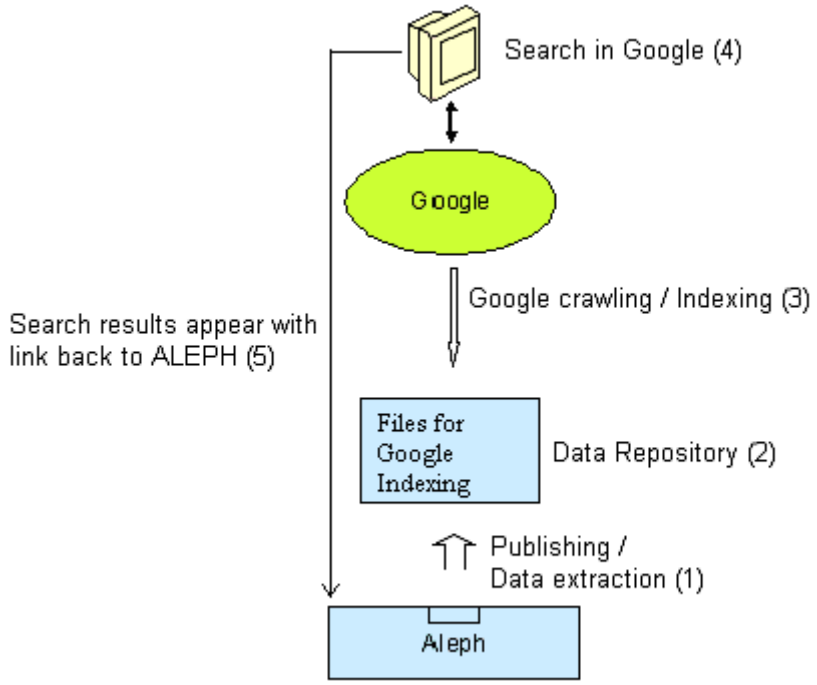
1 Purpose and Scope

ALEPH's Publishing Platform enables sites to extract records from the ALEPH catalog for various publishing purposes. The Publishing Platform is described in the document, *ALEPH Publishing Mechanism*, which is found in the Ex Libris Documentation Center.

One of the options that the ALPEH Publishing Platform offers is to publish the ALEPH bibliographic records for search and retrieval using the Google search engine. This document describes the flow and setup needed for successful Google publishing.

2 Architecture

The following is a schematic description of how ALEPH data is published for Google searches. The blue parts are performed in ALEPH. The yellow part is performed in Google.



1. **Publishing information from ALEPH –** ALEPH extracts the records that are required for publishing, and saves them in a format that can be further published for Google's use. This stage is done by the ALEPH Publishing Platform, and is described in detail in the document, *ALEPH Publishing Mechanism*.

2. **Creating files for Google Indexing** – The records that have been extracted and published go through the following processing:
    a. The records are moved from the ALEPH server to a Web server where they can be publicized for Google's use.
    b. The records are further structured to enable indexing by Google's crawler.

3. **Web search engine crawling** – This is the indexing process that is done by the Google crawler.

4. **Google searching –** A search is performed using the Google search engine. The results appear with a link to the ALEPH catalog.

5. **Link to ALEPH** – The user requests to view the found record. The record is displayed in the ALEPH Web OPAC.

The following chapters describe the special workflows and setups that are required in each of the above listed steps of the Google Publishing process.

3 Publishing Information from ALEPH

This section describes the special configurations that are required in the Publishing Platform to meet the Google requirements. The configurations described below effect the Initial Publishing Process service (publish-04), the Update Publishing Data (ue_21) daemon, and the Z00P records which are created by these services.

3.1 Defining the Set

Z00P records are created by the Initial Publishing Process (publish-04) service or by the Update Publishing Data (ue_21) daemon. For the Z00P records to be useful for Google publishing they must be in HTML format. For this purpose, the tab_publish Publishing Set (column 1) must be prefixed by WEB_PUB, and the tab_publish Format (column 5) must be hardcoded to HTML. Lines such as the following must therefore be defined in the tab_publish table:

```
!         1                    2        3   4          5
!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!-!-!!!!!!-!!!!!!!!!!!!!!!
WEB_PUB1               MATH             N             HTML
WEB_PUB2               PHYS             N             HTML
```

3.2 UE_21 Performance

The performance of ue_21 by can be improved by setting the aleph_start/ prof_library variable: `num_ue_21_processes`. This variable enables you to divide the running of the job into several processes. The variable can be set in aleph_start or in the prof_library file of the publishing library. Setting the variable in $alephe_root/aleph_start or aleph_start.private affects all of the publishing libraries. Setting the variable in $data_root/prof_library of the publishing library affects only this library.

For more information about the tab_publish table, refer to the document, *Aleph Publishing Mechanism*.

3.3 Formatting the Published Record

After the Publishing Set has been defined as HTML format, as described in the previous chapter, additional configuration is required so that the HTML record is properly formatted. This is done by configuring the following tables in the data_tab directory of the BIB library:

- tab_font_publish
- tab_doc_publish

3.3.1 tab_font_publish

This table is defined in the data_tab directory of the BIB library. It defines font styles for the Z00P HTML records. When the Publishing Platform creates the HTML records in the Z00P records, the fonts are concatenated according to the font-family information of the <style> tag.

Consider the following example:

tab_font_publish is set to:

```
!        1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
Bitstream Cyberbit
Arial Unicode MS
Arial
```

The <style> tag is created as:

"<style type=""text/css"">body {font-family: 'Bitstream Cyberbit', 'Arial Unicode MS', 'Arial';} </style> ".

3.3.2 tab_doc_publish

An HTML page that is indexed by Google must include the following parts:

- Title section.
- Meta description section
- Meta keywords section
- Body section
  Within the body section, different weights may be assigned to titles or to bolded text.

In addition, when Google indexes a document, different weights are given to different parts of the document. For example, the following indexing considerations are used many times by web search engines:

- More weight is given to the words that are found within headings (<h1></h1>).
- More weight is given to words in bold.

The tab_doc_publish table is defined in the data_tab directory of the BIB library. The table defines:

1. What fields of the extracted document are used in the above listed parts of the indexed HTML page.
2. What weight each field is given when indexed.
3. How the field are edited in the HTML document.

The table is made up of two columns:

- Column 1 – Type of the field
  Four types are configurable, corresponding to the four parts of the HTML document:
  - **T – Title section –** Only a single T line is allowed.
  - **D** – **Meta description section** – Only a single D line is allowed.

- **K** – **Meta keywords section** – Multiple K lines are allowed.
- **B** – **Body section** – Multiple B lines are allowed.

- Column 2 – What fields are extracted into the section and how the information is formatted. The parameters are separated by commas.

  The following are the allowed parameters for each field type:

  - **T** and **D** – A paragraph number from edit_paragraph, for example:

    ```
    !1        2
    !-!!!!!!!!!!!!!!!!!!!!>
    T 100
    D 100
    ```

    The edit_paragraph.lng and edit_field.lng tables define exactly which fields and subfields are to be included in the paragraph that is in the Title section (for T) or in the Meta description section (for D).

  - **K** – List of fields and sub-fields from which keywords must be created, and the filing code which are used when extracting the subfields. Each line includes a single field that is separated from the list of subfields by a comma. The list of subfields is separated from the filing code by another comma. For example:

    ```
    !1        2
    !-!!!!!!!!!!!!!!!!!!!!>
    K 7####,axyz,01
    K 6####,axyz,01
    ```

    Note that when keywords are created all built-in punctuation marks are dropped.

  - **B** – List of fields, edit_field identifiers from which the body section is created and the weight type that the field is assigned.

    Three weight types may be assigned:
    - Type 1 – Headings
    - Type 2 – Bold text
    - Type 3 – Default

    For example:
    ```
    !1        2
    !-!!!!!!!!!!!!!!!!!!!!>
    B 245##,D,1
    B 100##,D,1
    B 7####,E,2
    ```

    In this example field 245## will be extracted into the body section according to the edit_field.lng code (column 4) D. It is assigned the weight value 1, and therefore it is exported as a heading.

4 Creating Files for Google Indexing

This chapter describes actions and configurations that are required after the Publishing process itself has ended and the Create Tar File for ALEPH Published Records (publish-06) service has successfully created a tar file of the required Z00P records. These actions are further required in order to make the published records available for the Google crawler.

Special scripts must be executed in order to achieve the following:

- Move the tar file from the ALEPH server to the Web server directory which is open for external applications, such as the Google crawler. The move is done by using FTP.
- Extract the tar file and build a directory tree usable by the Google crawler.
- Index the output directory (build the index.html files).
- Delete the tar file in the ALEPH server and on the Web Server.

The following scripts, which are found in the ALEPH server, carry out the above listed actions:

- $aleph_proc/import_html_files
- $aleph_proc/build_web_tree.pl
- $aleph_proc/index_web_tree.pl

The scripts must be copied to the Web server, which is where they are executed.

Note: The Web server must be capable of running the cshell script and the Perl programs.

Running the scripts is done by executing import_html_files (which activates the other scripts). The script's parameters are:

a. **Target location** – the Web server location in which the directory tree is to be built. This location must be open for external browsers.
b. **Source location** – The path on which the tar file that publish-06 has created can be found. This path is on the ALEPH server.
c. **Source machine** – The IP of ALEPH server on which the tar file that publish-06 created can be found.
d. **User** – The user login of the ALEPH server.
e. **Password** – The password to the ALEPH server.

For example: csh –f import_html_files /apache/htdocs /exlibris/aleph/a18_1/export 10.1.235.14 tester testerpass

5 Link to ALEPH

The apache server must be configured to use a direct URL into the ALEPH system when access to the searched document is requested by the user.

This can be done by adding a line such as the following in apache/conf/httpd.conf

```
RewriteCond %{HTTP_USER_AGENT} !.*bot.*
RewriteRule ^/.*/.*/.*/.*/.*/([a-zA-Z0-9]+)\.([0-9]+)\.html$    \
```

http://<*ALEPH server*>/F/-?func=find-b&request=$2&find_code=SYS&local_base=$1

For example:

```
RewriteCond %{HTTP_USER_AGENT} !.*bot.*
RewriteRule ^/.*/.*/.*/.*/.*/([a-zA-Z0-9]+)\.([0-9]+)\.html$    \
http://catalogue.bl.uk/F?func=find-b&request=$2&find_code=SYS&local_base=$1
```